

# THE USE OF BAD PRIMES IN RATIONAL RECONSTRUCTION

JANKO BÖHM, WOLFRAM DECKER, CLAUS FIEKER, AND GERHARD PFISTER

**ABSTRACT.** A standard method for computing a rational number from its values modulo a collection of primes is to determine its value modulo the product of the primes via Chinese Remaindering, and then use Farey sequences for rational reconstruction. Successively enlarging the set of primes if needed, this method is guaranteed to work if we restrict ourselves to "good" primes. Depending on the particular application, however, there is often no efficient way of finding good primes. This note shows that in most situations, we can simply ignore this problem. In fact, we present an error tolerant algorithm for rational reconstruction. With regard to applications, we are particularly interested in the design of modular and, thus, parallel versions of algorithms in commutative algebra and algebraic geometry. Here, typically, the final result consists of one or several a priori unknown ideals which are found via constructions yielding the (reduced) Gröbner bases of the ideals.

## 1. INTRODUCTION

Rational reconstruction is a standard way of obtaining results in characteristic zero from results in characteristic  $p > 0$ . This is of particular use in the design of parallel algorithms and in situations where the growth of intermediate results matters. Classical applications are the computation of polynomial greatest common divisors (see [Wang 1981, Encarnación 1995]) and Gröbner bases (see [Arnold 2003, Idrees et al. 2011]). Here, the Gröbner bases algorithms start from an ideal already given. In contrast, more recent applications in commutative algebra and algebraic geometry (see, for example, [Böhm et al. 2011, Böhm et al. 2012]) require that we find an unknown ideal via a construction which computes the ideal by computing its (reduced) Gröbner basis. Here, for the purpose of modularization, we suppose that the construction applies to some given input data in characteristic zero as well as to "most" modular values of the input data. In such a situation, problems may arise in cases where the desired Gröbner basis in characteristic zero does not necessarily reduce to the Gröbner basis obtained in characteristic  $p$ . Usually, a first step to resolve these problems is to show that the "bad" primes  $p$  are "rare". Note, however, that the actual test of whether a given prime is bad (and should, hence, be discarded) may not be effective or may require expensive computations of invariants. Hence, a reconstruction algorithm which will return the correct result even in the presence of bad primes will be of great use. In this note, we describe such an algorithm. The algorithm will work whenever there are only finitely many bad primes.

---

*Date:* July 9, 2012.

*Key words and phrases.* rational reconstruction, Farey map.

To begin, in Section 2, we recall the classical approach to rational reconstruction which is based on the lifting of modular to rational results by computing Farey preimages via Euclidean division with remainder. In Section 3, to illustrate the need for error tolerant rational reconstruction, we discuss a general setup for applications in commutative algebra and algebraic geometry. Finally, in Section 4, we present the new lifting algorithm based on Gaussian reduction and discuss the resulting error tolerant reconstruction algorithm.

## 2. RECONSTRUCTION OF A SINGLE RATIONAL NUMBER

We describe the reconstruction of a single unknown number  $x \in \mathbb{Q}$ . In typical applications, this number will occur as a coefficient of some unknown vector or polynomial or element of a Gröbner basis. Also, frequently in this context, once the rational number (vector, polynomial, Gröbner basis) has been found, it is comparably easy to verify the correctness of the result.

We use the following notation: Given an integer  $N \geq 2$  and a number  $x = a/b \in \mathbb{Q}$  with  $\gcd(a, b) = 1$  and  $\gcd(b, N) = 1$ , the *value of  $x$  modulo  $N$*  is

$$x_N := \left(\frac{a}{b}\right)_N := (a + N\mathbb{Z})(b + N\mathbb{Z})^{-1} \in \mathbb{Z}/N\mathbb{Z}.$$

We also write  $x \equiv r \pmod{N}$  if  $r \in \mathbb{Z}$  represents  $x_N$ .

In what follows, we suppose that in the context of some application, we are given an algorithm which computes the value of the unknown number  $x \in \mathbb{Q}$  modulo any prime  $p$ , possibly rejecting the prime. For reference purposes, we formulate this in the black box type Algorithm 1.

---

### Algorithm 1 Black Box Algorithm $x \pmod{p}$

---

**Input:** A prime number  $p$ .

**Output:** **false** or an integer  $0 \leq s \leq p-1$  such that  $x \equiv s \pmod{p}$ .

*Assumption:* There are only finitely many primes  $p$  where the return value is **false**.

---

Once the values of  $x$  modulo the primes in a sufficiently large set of primes  $\mathcal{P}$  have been computed, we may find  $x$  via a lifting procedure consisting of two steps: First, use Chinese remaindering to obtain the value of  $x$  modulo the product  $N := \prod_{p \in \mathcal{P}} p$ . Second, compute the preimage of this value under the *Farey rational map* which is defined as follows.

For an integer  $B > 0$ , set

$$F_B = \left\{ \frac{a}{b} \in \mathbb{Q} \mid \gcd(a, b) = 1, 0 \leq a \leq B, 0 < |b| \leq B \right\},$$

and for  $m \in \mathbb{Z}/N\mathbb{Z}$ , let

$$\mathbb{Q}_{N,m} = \left\{ \frac{a}{b} \in \mathbb{Q} \mid \gcd(a, b) = 1, \gcd(b, N) = 1, \left(\frac{a}{b}\right)_N = m \right\}$$

be the set of rational numbers whose value modulo  $N$  is  $m$ . Then  $\mathbb{Q}_N = \bigcup_{m=0}^{N-1} \mathbb{Q}_{N,m}$  is a subring of  $\mathbb{Q}$  with identity. If  $B$  is an integer with  $B \leq \sqrt{(N-1)/2}$ , then the *Farey map*

$$\varphi_{B,N} : F_B \cap \mathbb{Q}_N \rightarrow \mathbb{Z}/N\mathbb{Z}, \quad \frac{a}{b} \mapsto \left(\frac{a}{b}\right)_N,$$

is well-defined and injective (but typically not surjective). To obtain the injective map with the largest possible image for a given  $N$ , we tacitly suppose in what follows that  $B$  is chosen as large as possible for  $N$ .

Algorithm 2 below will return  $\varphi_{B,N}^{-1}(\bar{r})$  if  $\bar{r}$  is in the image of the Farey map, and **false** otherwise (see, for example, [Kornerup and Gregory 1983, Wang 1981, Wang et al. 1982]).

---

**Algorithm 2** Farey Preimage

---

**Input:** Integers  $N \geq 2$  and  $0 \leq r \leq N - 1$ .

**Output:** **false** or a rational number  $a/b$  with  $\gcd(a, b) = 1$ ,  $\gcd(b, N) = 1$ ,

$a/b \equiv r \pmod{N}$ ,  $0 \leq a \leq \sqrt{(N-1)/2}$ ,  $0 < |b| \leq \sqrt{(N-1)/2}$ .

1:  $(a_0, b_0) := (N, 0)$ ,  $(a_1, b_1) := (r, 1)$ ,  $i := -1$

2: **while**  $2a_{i+2}^2 \geq N - 1$  **do**

3:    $i := i + 1$

4:   divide  $a_i$  by  $a_{i+1}$  to find  $q_i, a_{i+2}, b_{i+2}$  such that

$$(a_i, b_i) = q_i(a_{i+1}, b_{i+1}) + (a_{i+2}, b_{i+2})$$

and  $0 \leq a_{i+2} < a_{i+1}$

5: **if**  $2b_{i+2}^2 < N - 1$  **and**  $\gcd(a_{i+2}, b_{i+2}) = 1$  **then**

6:   **return**  $a_{i+2}/b_{i+2}$

7: **return false**

---

*Remark 2.1.* As pointed out in [Collins et al. 1994], dropping the requirement  $\gcd(a_{i+2}, b_{i+2}) = 1$  may lead to an invalid result: For  $N = 12$  and  $r = 5$ , the algorithm would return  $2/-2$ , but  $-2$  and  $12$  are not coprime. Note, however, that  $2 \equiv (-2) \cdot 5 \pmod{12}$  and, thus,  $1 \equiv (-1) \cdot 5 \pmod{6}$ .

Summing up, we get the classical reconstruction Algorithm 3.

---

**Algorithm 3** Reconstruction of a Rational Number

---

**Input:** Algorithm 1 and a way to verify that a computed number equals  $x$ .

**Output:**  $x$

1:  $N := 1$ ,  $r := 0$

2:  $p := 2$

3: **loop**

4:   let  $s$  be the return value of Algorithm 1 applied to  $p$

5:   **if**  $s = \mathbf{false}$  **then**

6:     continue with step 13

7:   find  $1 = eN + fp$  and set  $r := rfp + seN$ ,  $N := Np$

8:   let  $y$  be the return value of Algorithm 2 applied to  $N$  and  $r$

9:   **if**  $y = \mathbf{false}$  **then**

10:    continue with step 13

11:   **if**  $y = x$  **then**

12:    **return**  $y$

13:    $p := \text{NextPrime}(p)$

---

We remind the reader that our setup in this section is somewhat special in that we suppose that our Black Box Algorithm 1 either returns **false** or

a correct answer. For most applications, however, there exist primes  $p$  which are bad in the sense that the algorithm under consideration returns a wrong answer modulo  $p$ . This can, for example, happen in linear algebra: Suppose, we are given a matrix  $M \in \mathbb{Z}^{n \times n}$  with  $\dim \ker M = 1$ . For each prime  $p$ , let  $M_p$  be the reduction of  $M$  modulo  $p$ . Then, in order to compute (a basis vector of)  $\ker M$ , we may compute  $\ker M_p$  for suitable primes  $p$ , and then apply the strategy outlined above to the coefficients of the basis vectors. As long as always  $\dim \ker M_p = 1$ , the lifting will work if the computed kernel vectors are conveniently normalized to make them unique. On the other hand, if we encounter a bad prime for  $M$ , that is, a prime  $p$  with  $\dim \ker M_p > 1$ , we typically find a random kernel vector modulo  $p$ . This is not a problem since primes which are bad for  $M$  can be easily detected by checking the rank of  $M_p$ . For the applications we have in mind, however, detecting bad primes may not be feasible. In this note, we show that if there are only finitely many bad primes, they can just be ignored. More precisely, we show that in Algorithm 3, we may call the black box type Algorithm 4 below instead of Algorithm 1, provided we call the lifting Algorithm 6 from Section 4 instead of Algorithm 2.

---

**Algorithm 4** Black Box Algorithm  $x \bmod p$

---

**Input:** A prime number  $p$ .

**Output:** **false** or an integer  $0 \leq s \leq p - 1$ .

*Assumption:* There are only finitely many primes  $p$  where either the return value is **false** or  $x \not\equiv s \bmod p$ .

---

### 3. A SETUP FOR APPLICATIONS IN ALGEBRA AND GEOMETRY

As a motivation for the error tolerant version of rational reconstruction presented in the next section, we use this section to discuss a general computational setup for applications in commutative algebra and algebraic geometry which requires error tolerance. A setup of this type occurs, for example, when computing normalization or when computing adjoint curves. See [Böhm et al. 2011, Böhm et al. 2012]) and Example 3.9 below.

To begin, fix a global monomial ordering  $>$  on the semigroup of monomials in the variables  $X = \{X_1, \dots, X_n\}$ . Consider the polynomial rings  $W = \mathbb{Q}[X]$  and, given an integer  $N \geq 2$ ,  $W_N = (\mathbb{Z}/N\mathbb{Z})[X]$ . If  $T \subset W$  or  $T \subset W_N$  is a set of polynomials, then denote by  $\text{LM}(T) := \{\text{LM}(f) \mid f \in T\}$  its set of leading monomials. If  $f \in W$  is a polynomial such that  $N$  is coprime to any denominator of a coefficient of  $f$ , then its *reduction modulo  $N$*  is the polynomial  $f_N \in W_N$  obtained by mapping each coefficient  $x$  of  $f$  to  $x_N$  as described in the previous section. If  $H = \{h_1, \dots, h_t\} \subset W$  is a Gröbner basis such that  $N$  is coprime to any denominator in any  $h_i$ , set  $H_N = \{(h_1)_N, \dots, (h_t)_N\}$ . If  $J \subseteq W$  is any ideal, its *reduction modulo  $N$*  is the ideal

$$J_N = \langle f_N \mid f \in J \cap \mathbb{Z}[X] \rangle \subseteq W_N.$$

NOTATION: From now on, let  $I \subset W$  be a fixed ideal.

*Remark 3.1.* For practical purposes,  $I$  is given by a set of generators. Fix one such set  $f_1, \dots, f_r$ . Then the reduction of  $I$  modulo a prime  $p$  can be realized via the equality

$$I_p = \langle (f_1)_p, \dots, (f_r)_p \rangle \subseteq W_p$$

which holds for all but finitely many primes  $p$ . When performing the modular Algorithm 5 described below, we reject a prime  $p$  if one of the  $(f_i)_p$  is not defined. Otherwise, we work with the ideal on the right hand side instead of  $I_p$ . This does not cause problems: The finitely many bad primes where  $\langle (f_1)_p, \dots, (f_r)_p \rangle$  differs from  $I_p$  will not influence the result if we apply error tolerant rational reconstruction.

In what follows, we suppose that we are given a construction which associates to  $I$  a uniquely determined ideal  $U(0) \subset W$ , and to each reduction  $I_p$ , with  $p$  a prime number, a uniquely determined ideal  $U(p) \subset W_p$ , where we make the following assumption:

ASSUMPTION: We ask that  $U(0)_p = U(p)$  for all but finitely many  $p$ .

We write  $G(0)$  for the uniquely determined reduced Gröbner basis of  $U(0)$ , and  $G(p)$  for that of  $U(p)$ . In the applications we have in mind, we wish to construct the unknown ideal  $U(0)$  from a collection of its characteristic  $p$  counterparts  $U(p)$ . Technically, given a finite set of primes  $\mathcal{P}$ , we wish to construct  $G(0)$  by computing the  $G(p)$ ,  $p \in \mathcal{P}$ , and lifting the  $G(p)$  coefficientwise to characteristic zero. Here, to identify Gröbner basis elements corresponding to each other, we require that  $\text{LM}(G(p)) = \text{LM}(G(q))$  for all  $p, q \in \mathcal{P}$ .

**Definition 3.2.** With notation as above, we define:

- (1) A prime number  $p$  is called *lucky* if the following hold:
  - (a)  $U(0)_p = U(p)$  and
  - (b)  $\text{LM}(G(0)) = \text{LM}(G(p))$ .
 Otherwise  $p$  is called *unlucky*.
- (2) A finite set  $\mathcal{P}$  of lucky primes is called *sufficiently large* if

$$\prod_{p \in \mathcal{P}} p > \max \left\{ 2 \cdot |c|^2 \mid \begin{array}{l} c \text{ a denominator or numerator} \\ \text{of a coefficient occurring in } G(0) \end{array} \right\}.$$

*Remark 3.3.* A modular algorithm for the fundamental task of computing Gröbner bases is presented in [Arnold 2003] and [Idrees et al. 2011]. In contrast to our situation here, where we wish to find the ideal  $U(0)$  by computing its reduced Gröbner basis  $G(0)$ , Arnold's algorithm starts from an ideal which is already given. If  $p$  is a prime number,  $J \subset W$  is an ideal,  $H(0)$  is the reduced Gröbner basis of  $J$ , and  $H(p)$  is the reduced Gröbner basis of  $J_p$ , then  $p$  is lucky for  $J$  in the sense of Arnold if  $\text{LM}(H(0)) = \text{LM}(H(p))$ . It is shown in [Arnold 2003, Thm. 5.12 and 6.2] that if  $J$  is homogeneous and  $p$  is lucky for  $J$  in this sense, then  $H(0)_p$  is well-defined and equal to  $H(p)$ . Furthermore, by [Arnold 2003, Cor. 5.4 and Thm. 5.13], all but finitely many primes are Arnold-lucky for a homogeneous  $J$ . Using weighted homogenization as in the proof of [Idrees et al. 2011, Thm. 2.4], one shows that these results also hold true in the non-homogeneous setup.

With respect to our notation of lucky, we have:

**Lemma 3.4.** *The set of unlucky primes is finite.*

*Proof.* By our general assumption,  $U(0)_p = U(p)$  for all but finitely many primes  $p$ . Given a prime  $p$  such that  $U(0)_p = U(p)$ , we have  $\text{LM}(G(0)) = \text{LM}(G(p))$  if  $p$  does not divide any denominator of any coefficient of any polynomial occurring in Buchberger's Gröbner basis test for  $G(0)$ . The result follows.  $\square$

**Lemma 3.5.** *If  $\mathcal{P}$  is a sufficiently large set of lucky primes, then the reduced Gröbner bases  $G(p)$ ,  $p \in \mathcal{P}$ , lift to the reduced Gröbner basis  $G(0)$ .*

*Proof.* If  $p$  is lucky, then  $p$  is Arnold-lucky for  $U(0)$ . Hence, as remarked above,  $G(0)_p = G(p)$ . Since  $\mathcal{P}$  is sufficiently large, the coefficients of the Chinese remainder lift  $G(N)$ ,  $N = \prod_{p \in \mathcal{P}}$ , are in the image of the Farey map. Since this map is injective, the lift of  $G(N)$  to characteristic zero coincides with  $G(0)$ .  $\square$

From a theoretical point of view, the idea of finding  $G(0)$  is now as follows: Consider a sufficiently large set  $\mathcal{P}$  of lucky primes, compute the reduced Gröbner bases  $G(p)$ ,  $p \in \mathcal{P}$ , and lift the results to  $G(0)$  as described above.

From a practical point of view, we face the problem that the defining conditions of lucky and sufficiently large in Definition 3.2 cannot be tested a priori. With regard to condition (1a), for instance, we compute  $G(p)$  and, thus,  $U(p)$  on our way, but  $U(0)_p$  is only known to us after  $G(0)$  and, thus,  $U(0)$  have been found. As in Remark 3.1, this is not a problem: Finitely many bad primes leading to an ideal  $U(p)$  different from  $U(0)_p$  will not influence our final result if we apply error tolerant rational reconstruction. Condition (1b), on the other hand, is crucial since we use the leading monomials to identify Gröbner basis elements corresponding to each other. We therefore proceed in the following randomized way. First, fix an integer  $t \geq 1$  and choose a set of  $t$  primes  $\mathcal{P}$  at random. Second, compute  $\mathcal{GP} = \{G(p) \mid p \in \mathcal{P}\}$ , and use the following test to modify  $\mathcal{P}$  so that all primes in  $\mathcal{P}$  satisfy condition (1b) with high probability:

**DELETEUNLUCKYPRIMES:** *Define an equivalence relation on  $\mathcal{P}$  by setting  $p \sim q : \iff \text{LM}(G(p)) = \text{LM}(G(q))$ . Then replace  $\mathcal{P}$  by an equivalence class of largest cardinality<sup>1</sup>, and change  $\mathcal{GP}$  accordingly.*

Only now, we lift the Gröbner bases in  $\mathcal{GP}$  to a set of polynomials  $G \subseteq W$ . Since we do not know whether all primes in the chosen equivalence class are indeed lucky and whether the class is sufficiently large, a final verification in characteristic zero is needed. As this may be expensive, especially if  $G \neq G(0)$ , we first perform a test in positive characteristic:

**PTEST:** *Randomly choose a prime number  $p \notin \mathcal{P}$  such that  $p$  does not divide the numerator and denominator of any coefficient occurring in a polynomial in  $G$  or  $\{f_1, \dots, f_r\}$ . Return true if  $G_p = G(p)$ , and false otherwise.*

If PTEST returns **false**, then  $\mathcal{P}$  is not sufficiently large, or not all primes in  $\mathcal{P}$  are lucky (or the extra prime chosen in PTEST is unlucky). In this

---

<sup>1</sup>When computing the cardinality, take Remark 3.6 below into account.

case, we enlarge the set  $\mathcal{P}$  by  $t$  primes not used so far, and repeat the whole process. If `PTEST` returns true, however, then most likely  $G = G(0)$ . It makes, hence, sense to verify the result over the rationals. If the verification fails, we enlarge  $\mathcal{P}$  and repeat the process.

We summarize this approach in Algorithm 5.

---

**Algorithm 5** Reconstruction of an Ideal

---

**Input:** An algorithm to compute  $G(p)$  from  $I_p$ , for each prime  $p$ , and a way of verifying that a computed Gröbner basis equals  $G(0)$ .

**Output:** The Gröbner basis  $G(0)$ .

```

1: choose a list  $\mathcal{P}$  of random primes
2:  $\mathcal{GP} = \emptyset$ 
3: loop
4:   for  $p \in \mathcal{P}$  do
5:     compute  $G(p) \subseteq W_p$ 
6:      $\mathcal{GP} = \mathcal{GP} \cup \{G(p)\}$ 
7:    $(\mathcal{P}, \mathcal{GP}) = \text{DELETEUNLUCKYPRIMES}(\mathcal{P}, \mathcal{GP})$ 
8:   lift  $\mathcal{GP}$  to  $G \subseteq W$  via Chinese remaindering and Algorithm 6 below
9:   if the lifting succeeds and PTEST( $I, G, \mathcal{P}$ ) then
10:    if  $G = G(0)$  then
11:      return  $G$ 
12:   enlarge  $\mathcal{P}$  with primes not used so far

```

---

*Remark 3.6.* If Algorithm 5 requires more than one round of the loop, the cardinality count in `DELETEUNLUCKYPRIMES` has to be done with some care: count all previous elements of  $\mathcal{P}$  as just one element. Otherwise, though highly unlikely in practical terms, it may happen that only unlucky primes are accumulated.

*Remark 3.7.* Our lifting process works since reduced Gröbner bases are uniquely determined. In practical terms, however, there is often no need to reduce the Gröbner bases involved: it is only required that the construction associating the Gröbner bases to  $I$  and its reductions yields uniquely determined results.

*Remark 3.8.* We may allow that the computation of  $G(p)$  is not feasible for finitely many primes  $p$ . In this case, the respective primes will be rejected.

*Example 3.9.* If  $K$  is any field, and  $I \subset K[X]$  is a prime ideal, the *normalization*  $\overline{A}$  of the domain  $A = K[X]/I$  is the integral closure of  $A$  in its field of fractions  $\mathbb{Q}(A)$ . If  $K$  is perfect, the normalization algorithm given in [Greuel et al. 2010] will find a “valid denominator”  $d \in A$  and an ideal  $U \subset A$  such that  $\frac{1}{d}U = \overline{A} \subset \mathbb{Q}(A)$ . In fact,  $U$  is uniquely determined if we fix  $d$ . In practical terms,  $d$  and  $U$  are a polynomial and an ideal in  $K[X]$ , respectively. If  $K = \mathbb{Q}$  and  $p$  is a prime number, it may happen that  $I_p$  is not a prime ideal, that  $d_p$  is not defined, or that  $d_p$  is not a valid denominator. See [Böhm et al. 2011] for the modular normalization algorithm.

## 4. RECONSTRUCTION WITH BAD PRIMES

In order to show that a reconstruction scheme as in Algorithm 5 can be used even in the presence of bad primes, we turn rational reconstruction into a lattice problem.

To begin with, given an integer  $N \geq 2$ , we define the subset  $C_N \subseteq \mathbb{Z}/N\mathbb{Z}$  of elements applied to which Algorithm 6 below will return a rational number (and not **false**). Let  $C_N$  be the set of all  $\bar{r} \in \mathbb{Z}/N\mathbb{Z}$  such that there are integers  $u, v \in \mathbb{Z}$  with  $u \geq 0$ ,  $v \neq 0$ , and  $\gcd(u, v) = 1$  which satisfy the following condition:

$$\begin{aligned} &\text{there is an integer } q \geq 1 \text{ with } q|N \text{ and such that} \\ &u^2 + v^2 < \frac{N}{q^2} \quad \text{and} \quad u \equiv vr \pmod{\frac{N}{q}}. \end{aligned} \tag{1}$$

In Lemma 4.2 below, we will prove that the rational number  $\frac{u}{v} = \frac{uq}{vq} \in \mathbb{Q}$  is uniquely determined by Condition (1). Hence, we have a well-defined map

$$\psi_N : C_N \rightarrow \mathbb{Q}.$$

Note that the image of the Farey map  $\varphi_{B,N}$ , with  $B = \lfloor \sqrt{(N-1)/2} \rfloor$ , is contained in  $C_N$ : If  $\bar{r} \in \text{im}(\varphi_{B,N})$ , then  $\varphi_{B,N}^{-1}(\bar{r})$  satisfies Condition (1) with  $q = 1$ . Furthermore,  $\varphi_{B,N}^{-1}(\bar{r}) = \psi_n(\bar{r})$ .

Typically, the inclusion  $\text{im}(\varphi_{B,N}) \subseteq C_N$  is strict:

*Example 4.1.* For  $N = 2 \cdot 13$ , we have  $B = 3$ , hence

$$\text{im}(\varphi_{B,N}) = \{\bar{0}, \bar{1}, \bar{2}, \bar{3}, \bar{8}, \bar{9}, \bar{17}, \bar{18}, \bar{23}, \bar{24}, \bar{25}\},$$

and the rational numbers which can be reconstructed by Algorithm 2 are the elements of

$$F_B \cap \mathbb{Q}_N = \left\{ 0, \pm 1, \pm 2, \pm 3, \pm \frac{1}{3}, \pm \frac{2}{3} \right\}.$$

On the other hand,

$$C_N = \{\bar{r} \mid 0 \leq r \leq 25, r \neq 5, 21\},$$

and Algorithm 6 will reconstruct the rational numbers in

$$\psi_N(C_N) = \left\{ 0, \pm 1, \pm 2, \pm 3, \pm 4, \pm \frac{1}{2}, \pm \frac{1}{3}, \pm \frac{2}{3}, \pm \frac{4}{3} \right\}.$$

Note that the denominator of  $\frac{1}{2} = \psi_N(7) = \psi_N(20)$  is not coprime to  $N$ . In both cases,  $q = 2$ : We have  $1 \equiv 2 \cdot 7 \pmod{13}$  and  $1 \equiv 2 \cdot 20 \pmod{13}$ .

Now, fix  $0 \leq r \leq N-1$  such that  $\bar{r} \in C_N$ , and consider the lattice  $\Lambda = \Lambda_{N,r} := \langle (N, 0), (r, 1) \rangle$  of discriminant  $N$ . Let  $u, v, q$  correspond to  $\bar{r}$  as in Condition (1). Then  $(uq, vq) \in \Lambda_{N,r}$ . Hence, the first minimum  $m_1(\Lambda)$  of  $\Lambda$  satisfies  $m_1(\Lambda) \leq q^2(u^2 + v^2)$ .

**Lemma 4.2.** *With notation as above, all  $(x, y) \in \Lambda$  with  $x^2 + y^2 < N$  are collinear. That is, they define the same rational number  $x/y$ .*

*Proof.* Let  $\lambda = (x, y)$ ,  $\mu = (c, d) \in \Lambda$  be vectors with  $x^2 + y^2, c^2 + d^2 < N$ . Then  $y\mu - d\lambda = (yc - xd, 0) \in \Lambda$ , so  $N|(yc - xd)$ . Since  $|yc - xd| < N$  by Cauchy-Schwarz, we get  $yc = xd$ , as claimed.  $\square$



Next, consider integers  $N', M \geq 2$ , with  $\gcd(M, N') = 1$ , and such that  $N = N'M$ . Let  $a \geq 0, b \neq 0$  be integers such that  $\gcd(b, N') = 1$ , and let  $a \equiv bs \pmod{N'}$ , with  $0 \leq s \leq N' - 1$ . Let  $0 \leq t \leq M - 1$  be another integer, and let  $0 \leq r \leq N - 1$  be the Chinese remainder lift satisfying  $r \equiv s \pmod{N'}$  and  $r \equiv t \pmod{M}$ . In practical applications, we think of  $N'$  and  $M$  as the product of good and bad primes, respectively. By the following lemma, Algorithm 6 below applied to  $N$  and  $r$  will return  $a/b$  independently of the possibly “wrong result”  $t$ , provided that  $M \ll N'$ .

**Lemma 4.3.** *With notation as above, suppose that  $(a^2 + b^2)M < N'$ . Then, for all  $(x, y) \in \Lambda = \langle (N, 0), (r, 1) \rangle$  with  $(x^2 + y^2) < N$ , we have  $x/y = a/b$ . Furthermore, if  $\gcd(a, b) = 1$  and  $(x, y)$  is a shortest nonzero vector in  $\Lambda$ , we also have  $\gcd(x, y) | M$ .*

*Proof.* From  $a \equiv bs \pmod{N'}$ , we get  $a - bs = k_1 N'$  for some  $k_1$ . Moreover,  $s \equiv r \pmod{N'}$  gives  $r = s + k_2 N'$ . Now  $(aM, bM) - bM(r, 1) = (aM - brM, 0)$  and  $aM - brM = M(a - br) = M(a - b(s + k_2 N')) = M(a - bs) - k_2 bN = k_1 N - k_2 bN$ , thus  $(aM, bM) \in \Lambda$ . Since  $(a^2 + b^2)M < N'$ , Lemma 4.2 gives  $a/b = aM/bM = x/y$  for all  $(x, y) \in \Lambda$  such that  $(x^2 + y^2) < N$ .

For the second statement, write  $A := (aM, bM)$  and  $X := (x, y)$ . By Lemma 4.2, there is a  $\lambda = \frac{s}{r} \in \mathbb{Q}$ , with  $\gcd(s, t) = 1$ , and such that  $\lambda X = A$ . The Euclidean Algorithm gives integers  $e, f$  with  $er + sf = 1$ , hence

$$\frac{X}{r} = (er + sf) \frac{X}{r} = eX + fA \in \Lambda.$$

Since  $X$  is a shortest vector in the lattice, it follows that  $r = \pm 1$ , hence  $A = \pm sX$ . Since  $\gcd(a, b) = 1$ , we conclude that  $\gcd(x, y) | M$ .  $\square$

The use of this lemma is twofold. First, it allows us to ignore bad primes in the design of modular algorithms – as long as there are not too many bad primes. Second, factorizing the gcd of the components of a shortest lattice element can help us to identify bad primes. From a theoretical point of view, this makes the design of modular algorithms much simpler. From a practical point of view, we avoid expensive computation of invariants to eliminate bad primes.

Lemma 4.3 yields the correctness of both the new lifting Algorithm 6 and the resulting reconstruction Algorithm 3, calling black box Algorithm 4 instead of 1. In applications, the termination can be based either on the knowledge of a priori bounds on the height of  $x/y$  or on an a posteriori verification of the result. It should be mentioned that both methods are used: some problems allow for easy verification, while others yield good bounds.

*Remark 4.4.* Algorithm 6, which is just a special case of Gaussian reduction, will always find a shortest vector in the lattice generated by  $(N, 0)$  and  $(r, 1)$ . Moreover,  $b_i \neq 0$  for all  $i > 0$  since in every step the vector  $(a_i, b_i)$  gets shorter and, hence, cannot be equal to  $(N, 0)$ .

Even though Algorithm 6 looks more complicated than Algorithm 2, by [Nguyen et al. 2009, Section 3.3] the bit-complexity of both algorithms is the same:  $O(\log^2 N)$ .

**Algorithm 6** Error Tolerant Lifting**Input:** Integers  $N \geq 2$  and  $0 \leq r \leq N - 1$ .**Output:**  $\psi_N(\bar{r})$  if  $\bar{r} \in C_N$  and **false** otherwise.

1:  $(a_0, b_0) := (N, 0)$ ,  $(a_1, b_1) := (r, 1)$ ,  $i := -1$   
 2: **repeat**  
 3:    $i = i + 1$   
 4:   set

$$q_i = \left\lfloor \frac{a_i a_{i+1} + b_i b_{i+1}}{a_{i+1}^2 + b_{i+1}^2} + \frac{1}{2} \right\rfloor$$

5:   set  
        $(a_{i+2}, b_{i+2}) = (a_i, b_i) - q_i(a_{i+1}, b_{i+1})$   
 6: **until**  $a_{i+2}^2 + b_{i+2}^2 \geq a_{i+1}^2 + b_{i+1}^2$   
 7: **if**  $a_{i+1}^2 + b_{i+1}^2 < N$  **then**  
 8:   **return**  $a_{i+1}/b_{i+1}$   
 9: **else**  
 10: **return false**

*Example 4.5.* We reconstruct the rational number  $\frac{13}{12}$  using the modulus

$$N = 38885 = 5 \cdot 7 \cdot 11 \cdot 101.$$

With notation as above,  $a = 13$ ,  $b = 12$ ,  $r = 22684$ , and the Farey bound is

$$B = \left\lfloor \sqrt{(N-1)/2} \right\rfloor = 139.$$

Algorithm 2 applied to these data will correctly return  $\frac{13}{12}$ . Similarly for Algorithm 6 which generates the sequence

$$\begin{aligned} (38885, 0) &= 2 \cdot (22684, 1) + (-6483, -2), \\ (22684, 1) &= -3 \cdot (-6483, -2) + (3235, -5), \\ (-6483, -2) &= 2 \cdot (3235, -5) + (-13, -12), \\ (3235, -5) &= -134 \cdot (-13, -12) + (1493, -1613). \end{aligned}$$

Now, bad primes will enter the picture. Consider the Chinese remainder isomorphism

$$\varphi : \mathbb{Z}/5\mathbb{Z} \times \mathbb{Z}/7\mathbb{Z} \times \mathbb{Z}/11\mathbb{Z} \times \mathbb{Z}/101\mathbb{Z} \rightarrow \mathbb{Z}/38885\mathbb{Z}.$$

The preimage of  $\bar{r} = \left(\frac{13}{12}\right)_N$  is

$$\varphi^{-1}(\bar{r}) = (\bar{4}, \bar{4}, \bar{2}, \bar{60}).$$

That is,  $\bar{r}$  is the solution to the simultaneous congruences

$$\begin{aligned} x &\equiv 4 \pmod{5} \\ x &\equiv 4 \pmod{7} \\ x &\equiv 2 \pmod{11} \\ x &\equiv 60 \pmod{101}. \end{aligned}$$

If we make 101 a bad prime by changing the congruence  $x \equiv 60 \pmod{101}$  to  $x \equiv 61 \pmod{101}$ , we obtain

$$\varphi(\bar{4}, \bar{4}, \bar{2}, \bar{61}) = \overline{16524}.$$

Algorithm 6 then computes

$$\begin{aligned}(38885, 0) &= 2 \cdot (16524, 1) + (5837, -2), \\ (16524, 1) &= 3 \cdot (5837, -2) + (-987, 7), \\ (5837, -2) &= 6 \cdot (-987, 7) + (-85, 40), \\ (-987, 7) &= 10 \cdot (-85, 40) + (-137, 393).\end{aligned}$$

Hence the output  $\frac{85}{-40} = \frac{17}{8} \neq \frac{13}{12}$  is not the desired lift. The reason for this is that 101 is not small enough compared to its cofactor in  $N$ . Algorithm 2, on the other hand, returns **false** since the reduction process will also terminate with  $(85, -40)$  and these numbers are not coprime. Note that in a setup as in Section 3, the wrong result does not cause problems since it will be detected by PTEST. As a consequence, the set of primes in Algorithm 5 will be enlarged (without discarding previous results). Eventually, the good primes will outweigh the bad ones and Algorithm 6 will return the correct lift. It will even tell us which of the primes under consideration are bad primes. For example, replace the congruence  $x \equiv 4 \pmod{7}$  by  $x \equiv 2 \pmod{7}$ , so that

$$\varphi(\overline{4}, \overline{2}, \overline{2}, \overline{60}) = \overline{464}.$$

Then Algorithm 6 yields

$$\begin{aligned}(38885, 0) &= 84 \cdot (464, 1) + (-91, -84), \\ (464, 1) &= -3 \cdot (-91, -84) + (191, -251),\end{aligned}$$

and terminates with the correct lift

$$\frac{91}{84} = \frac{13}{12}.$$

Algorithm 2, on the other hand, will again return **false** since the reduction also terminates with  $(91, 84)$ .

Since

$$(13^2 + 12^2) \cdot 7 < 5 \cdot 11 \cdot 101,$$

Lemma 4.3 shows that 7 is small enough compared to its cofactor in  $N$ . Hence, the wrong result 2 modulo the bad prime 7 does not influence the result of the lift. In fact, all other possible congruences modulo 7 will lead to the same output. Note that  $\gcd(91, 85, N) = 7$  is the bad prime. Furthermore, note that in the example the lifting process involving the bad prime requires fewer steps than the process relying on good primes only.

## REFERENCES

- [Arnold 2003] Arnold, E. A.: *Modular algorithms for computing Gröbner bases*, J. Symb. Comput. 35, 403–419 (2003).
- [Böhm et al. 2011] Böhm, J.; Decker, W.; Laplagne, S.; Pfister, G.; Steenpaß, A.; Steidel, S.: *Parallel Algorithms for Normalization*. Preprint <http://arxiv.org/abs/1110.4299> (2011).
- [Böhm et al. 2012] Böhm, J.; Decker, W.; Laplagne, S.; Seelisch, F.: *Local to global algorithms for the Gorenstein adjoint ideal of a curve*. In preparation.
- [Collins et al. 1994] Collins, George E., Encarnación, Mark J.: *Efficient Rational Number Reconstruction*. J. Symb. Comput. 20, 287–297 (1995).
- [Encarnación 1995] Encarnación, Mark J.: *Computing GCDs of polynomials over algebraic number fields*. J. Symb. Comput. 20, 299–313 (1995).

- [Greuel et al. 2010] Greuel, G.-M.; Laplagne, S.; Seelisch, F.: *Normalization of rings*. J. Symb. Comput. 45, 887–901 (2010).
- [Idrees et al. 2011] Idrees, N.; Pfister, G.; Steidel, S.: *Parallelization of Modular Algorithms*. J. Symb. Comput. 46, 672–684 (2011).
- [Nguyen et al. 2009] Nguyen, P.Q.; Stehlé, D.: *Low-dimensional lattice basis reduction revisited*. ACM Transactions on Algorithms, Paper 46 (2009).
- [Kornerup and Gregory 1983] Kornerup, P.; Gregory, R. T.: *Mapping Integers and Hensel Codes onto Farey Fractions*. BIT Numerical Mathematics 23(1), 9–20 (1983).
- [Wang 1981] Wang, P. S.: *A  $p$ -adic algorithm for univariate partial fractions*. Proceedings SYMSAC '81, 212–217 (1981).
- [Wang et al. 1982] Wang, P. S.; Guy, M. J. T.; Davenport, J. H.:  *$P$ -adic reconstruction of rational numbers*. SIGSAM Bull, 2–3 (1982).

FACHBEREICH MATHEMATIK, TECHNICAL UNIVERSITY KAISERSLAUTERN, POSTFACH  
3049, 67653 KAISERSLAUTERN, GERMANY  
*E-mail address:* boehm@mathematik.uni-kl.de

FACHBEREICH MATHEMATIK, TECHNICAL UNIVERSITY KAISERSLAUTERN, POSTFACH  
3049, 67653 KAISERSLAUTERN, GERMANY  
*E-mail address:* decker@mathematik.uni-kl.de

FACHBEREICH MATHEMATIK, TECHNICAL UNIVERSITY KAISERSLAUTERN, POSTFACH  
3049, 67653 KAISERSLAUTERN, GERMANY  
*E-mail address:* fieker@mathematik.uni-kl.de

FACHBEREICH MATHEMATIK, TECHNICAL UNIVERSITY KAISERSLAUTERN, POSTFACH  
3049, 67653 KAISERSLAUTERN, GERMANY  
*E-mail address:* pfister@mathematik.uni-kl.de